

PR #25428 完整报告

sgl-project/sglang

[Fix] Probe speculative draft config via sglang get_config

合并时间: 2026-05-16 13:01

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25428>

执行摘要

- 一句话: 修复 Mistral-native EAGLE 草稿模型启动崩溃
- 推荐动作: 值得快速合并, 修复针对性强, 改动极小且经过充分测试 (GSM8K 精度回归测试通过、Gemma4 回归测试通过)。无需深入精读。

功能与动机

Mistral-native EAGLE drafts 启动时崩溃, 报错: `ValueError: Unrecognized model in mistralai/Mistral-Large-3-675B-Instruct-2512-Eagle. Should have amodel_typekey in its config.json...`。原因是这些检查点使用 `params.json` 而非 `config.json`, `transformers.AutoConfig` 无法解析。

实现拆解

1. 在 `python/sglang/srt/server_args.py` 的 `_resolve_speculative_algorithm_alias()` 函数中, 将配置探查加载器从 `transformers.AutoConfig.from_pretrained` 替换为 `sglang` 的 `get_config`。
2. `get_config` 来自 `sglang.srt.utils.hf_transformers_utils`, 能够路由 Mistral-native 检查点通过 Mistral 解析器。
3. 其余逻辑 (Gemma4 检测、EAGLE3 拒绝、算法提升) 完全不变。

关键文件:

- `python/sglang/srt/server_args.py` (模块 参数解析; 类别 source; 类型 dependency-wiring): 变更唯一文件: 将 `transformers.AutoConfig.from_pretrained` 替换为 `sglang` 内部 `get_config`, 以支持 Mistral-native 检查点解析。

关键符号: `_resolve_speculative_algorithm_alias`

关键源码片段

`python/sglang/srt/server_args.py`

变更唯一文件: 将 `transformers.AutoConfig.from_pretrained` 替换为 `sglang` 内部 `get_config`, 以支持 Mistral-native 检查点解析。

```
# 关键变更: 将 transformers.AutoConfig 替换为 sglang 内部的 get_config
# 以支持 Mistral-native 检查点 (使用 params.json 而非 config.json)
```

```

from sglang.srt.utils.hf_transformers_utils import get_config

def _resolve_speculative_algorithm_alias(
    speculative_algorithm: Optional[str],
    speculative_draft_model_path: Optional[str],
    trust_remote_code: bool = False,
) -> Optional[str]:
    """Resolve CLI speculative algorithm; NEXTN/EAGLE may become FROZEN_KV_MTP for
    Gemma4 assistant drafts."""

    is_gemma4_draft = False
    if speculative_draft_model_path:
        # 使用 get_config 而非 AutoConfig.from_pretrained
        # get_config 能解析 Mistral-native 检查点 (params.json)
        cfg = get_config(
            speculative_draft_model_path, trust_remote_code=trust_remote_code
        )
        is_gemma4_draft = "Gemma4AssistantForCausalLM" in (
            getattr(cfg, "architectures", None) or []
        )

    # 以下逻辑未变: EAGLE3 对 Gemma4 抛出错误, NEXTN/EAGLE 提升为 FROZEN_KV_MTP
    if speculative_algorithm == "EAGLE3" and is_gemma4_draft:
        raise ValueError(
            "Gemma4AssistantForCausalLM draft requires "
            "--speculative-algorithm NEXTN or EAGLE; EAGLE3 is "
            "not supported for this draft architecture."
        )

    if speculative_algorithm == "NEXTN" or speculative_algorithm == "EAGLE":
        if is_gemma4_draft:
            return "FROZEN_KV_MTP"
        return "EAGLE"

    return speculative_algorithm

```

评论区精华

无实质性 review 讨论；两位 reviewer (b8zhong 和 Fridge003) 均直接批准，未留下评论。变更逻辑清晰、改动极小，无需深入讨论。

- 暂无高价值评论线程

风险与影响

- 风险：核心路径变更：_resolve_speculative_algorithm_alias() 在推理启动时调用，影响所有使用 EAGLE 草稿模型的场景。但 get_config 是 sglang 内部函数，已适配 Mistral 和标准 transformers 模型，回归风险低。无安全或性能风险（一次启动开销）。

- 影响：影响用户：支持 Mistral-native EAGLE 草稿模型用户，消除启动崩溃。影响系统：启动时配置加载路径变更，运行时无影响。影响团队：无维护负担增加。影响程度：小范围修复。
- 风险标记：核心路径变更

关联脉络

- PR #24593 [diffusion] Generalize layerwise offload residency mixin to all components: 同仓库近期扩散模型 PR，与本 PR 无直接关联，但均为 sglang 项目中的功能修复与改进。